

**BOM REVISION CONTROL RESEARCH**

Regarding CAPA 18-14

**REVISION HISTORY**

|  |  |  |
| --- | --- | --- |
| **REV** | **REVISED BY** | **CHANGE DESCRIPTION** |
| 00 | Jordan Landry | Document created by Jordan Landry |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**ADDITIONAL COMMENTS**

|  |
| --- |
|  |

**Table Of Contents**

[1. SCOPE 4](#_Toc9433518)

[2. Findings and proposed solutions 4](#_Toc9433519)

[2.1 Store Excel BOM on Autodesk Vault 4](#_Toc9433520)

[2.2 Vault customization 5](#_Toc9433521)

[2.2.1 Vault data Standard (VDS) 5](#_Toc9433522)

[2.3 INVENTOR BOM 8](#_Toc9433523)

[2.3.1 OBSERVATIONS 8](#_Toc9433524)

[2.3.2 Summary 10](#_Toc9433525)

[3. final Solution 10](#_Toc9433526)

[4. Resources 13](#_Toc9433527)

# SCOPE

The purpose of this research task was to address the current CAPA regarding BOM revision control. The following is a list of guidelines that were established at the onset of the task:

1. Find a method to implement revision control for BOM. Explore all possible solutions and list the advantages and disadvantages of each. To promote buy-in from SM staff, the proposed solution must decrease current workload.
2. If there is to be an implementation of the proposed solution, it should be a gradual, non-disruptive implementation.
3. Find solution(s) that are scalable with future growth. Avoid any temporary “band-aid” fixes.

Taking the above into consideration, I sat down one-on-one with employees from Engineering and Drafting to gain insight and collect suggestions on how best to address the situation.

# Findings and proposed solutions

Store Excel BOM on Autodesk Vault

I will first propose the method of storing the BOM on Vault, as it addresses the CAPA quickly and directly. I did not stop at this method, as I feel strongly that Stella Maris has already out grown using Excel BOMs and should explore other BOM options.

Storing the Excel BOM in its related project folder on Vault will allow for immediate document and revision control. Below are the advantages and disadvantages of implementing this method.

**Advantages**

* It is possible to control the creation of a new Microsoft Office document within Vault. Before a new document is created, a template for the new document must be loaded into Vault. The style, structure, and properties of the template can be driven by Stella Maris. This would eliminate the issue of different looking or different style Excel BOMs (as well as other documents), and would standardize the formatting of the Excel document.
* Once a file is checked in, in order to make a change to the BOM, a user must first check out the file, thus initiating a new revision number when the file is checked back in. This would address the issue of revision control.
* Users with thin-client access are not allowed to check a file into vault, therefore if file was not already established as “read only” the user cannot override the checked-in file that sits in vault.
* This implementation would not cause disruption to current processes.

**Disadvantages**

* If a user who has capability of checking-in/out a file is editing the file, and another user is viewing the same file from thin-client in “read only” mode, it is possible that the thin-client user is looking at out-of-date data.  *A possible solution to this could be to add a “status” column to the Excel BOM. This can also present an issue as it would require constant status updating by SM personnel.*
* Currently requires Excel BOM to be checked in from Vault Office plugin. This uses up an Autodesk license or “seat”.
* The data that is being used to populate an Excel BOM (Item #, description, manufacturer, price, leadtime, etc.) is static and more often than not, does not match an item that is already within the Quick Books system.

**Summary**

As of now, Excel BOMs are shared between Engineering, Drafting and Project Management, either through email, or on the X: Drive in a project folder. During the initial design phase where Mechanical Engineering is in the process of creating an Excel BOM and doing a rough sketch flow diagram, the BOM goes through approximately four (4) iterations. This BOM again, travels back and forth between Engineering and Drafting through the server, or email. Although “read-only” security can be applied to the Excel BOM, there is a possibility that a previous revision of the Excel BOM can be accidentally submitted to Drafting or Project Management, resulting in incorrect information being used. While the proposed solution of locking the Excel BOM to a project folder on Vault would eliminate these issues, I thought it was necessary to dig deeper and see if there was a way to eliminate the need for an Excel BOM completely.

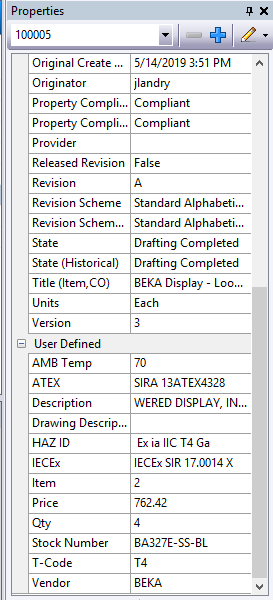
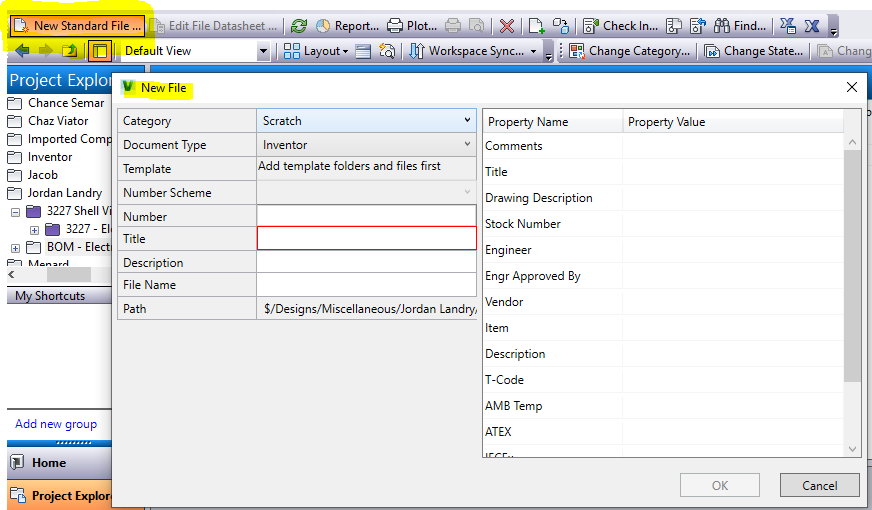
Vault customization

Before speaking about the next proposed solution, I will give a quick summary of applicable Vault customization tools and methods Stella Maris could employ.

#### Vault data Standard (VDS)

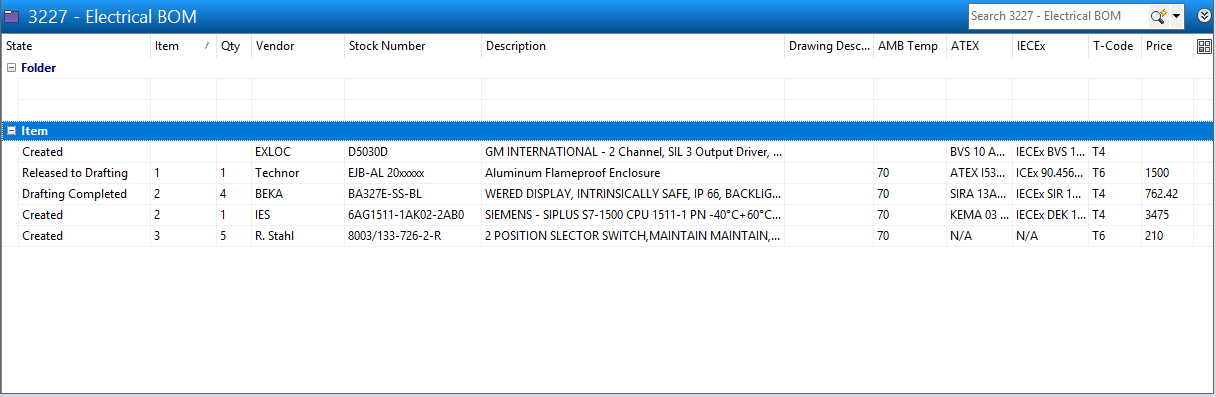
Through an extension called Vault Data Standard (VDS) it is possible to modify the User Interface (UI) of Vault without having to dig into the API. To best explain why VDS was created, it is helpful to understand how Vault operates without VDS. Data stored within Vault can be categorized as a Folder, File, Item or a Custom Object. From within each of these categories, sub-categories can be created, and each sub-category can be assigned its own properties. New custom properties can be quickly created and assigned to one or more sub-categories. The ability to create custom properties would allow Stella Maris to match those properties that exist on our Excel BOMs. For example, Hazardous Area markings for items within Electrical BOMs.

To keep this as brief as possible, we will assume that the categorization, and custom properties of Vault are already configured to meet Stella Maris needs. If a user was to create new data within Vault and apply relevant information to the property fields, the task of doing so is neither speedy nor user friendly.

****Vault Data Standard is a customizable User Interface (UI) that prompts the user enter all necessary information about the new data being created. The user can select data categories and fill property fields directly from this window. Without VDS, the Vault user would be doing the exact same thing but instead, doing it inside of the property window pane on the right side of the screen. Doing such is time consuming, tedious and prone to errors. Using VDS greatly speeds up the process of entering new data.

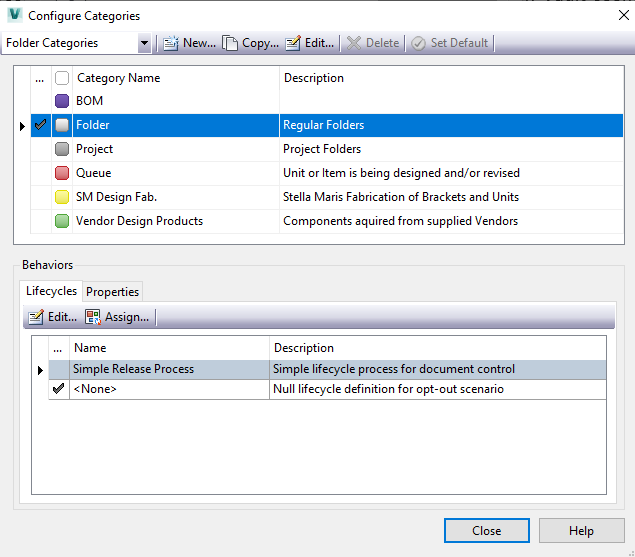
Vault Data Standard UI Window vs Property Grid

### SCRATCH ITEM BOM

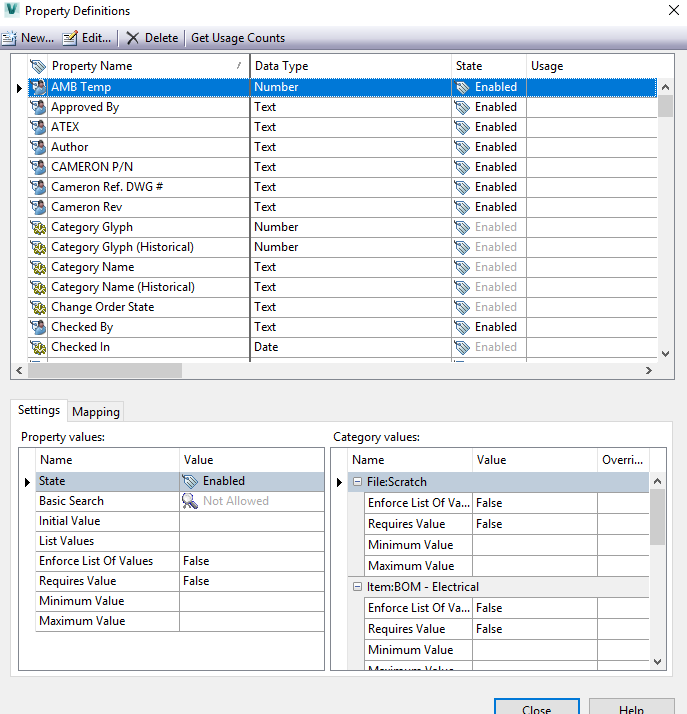
To address BOM revision control from a different approach, I explored the possibility of creating a “Scratch” BOM from scratch items. A scratch item is a blank item that can be created and configured to inherit user specified properties. The main purpose of a scratch item is to be able to add miscellaneous data to the Inventor auto-generated assembly BOM. Most of this type of data does not need or does not have a 2D or 3D drawing associated with it (i.e. paint, cabling, documents, etc.). A scratch item can be created either from the item master or from within the Inventor BOM. Upon creation, the scratch item inherits all relevant properties that are either pre-determined via categorial selection, or auto assigned through Inventor as the .iam assembly file is produced. This scratch BOM would allow Engineering to create for example, a folder titled “Electrical BOM” and populate it with product data using the scratch item option.

As Drafting would begin to create 2D and 3D drawings, they would look into this folder and find those items that were ready to be modeled. Engineering would create custom Life-Cycle definitions, and mark each BOM item as either “For Release to Drafting”, “Work in Progress”, or “For Release to Project Management”. This would allow the early release of Long Lead items, without releasing the entire BOM folder, which at this point in time would be incomplete. (Consider the BOM folder to be the equivalent of our current Excel BOM file)

Three issues arose with this strategy, one of which was that Vault Data Standard (and subsequent modification) was needed to make entering BOM data faster and more efficient. The second is that the information rich data that is being populated into this new style BOM is unavailable to be modified by those without an Autodesk license. Any change would require that the person modifying the BOM item to be checked out, and subsequently checked back in, creating a new revision of the item.

As of now my understanding is that Project Management takes the Excel BOM and manually enters the BOM data into Quick Books, along with more current and up-to-date information at the time of purchase. If a “scratch” BOM was to be utilized, a disconnect between Vault Data and Quick Books data would still exist. This would lead to the creation of duplicate and redundant information, unless the information was pushed to a central company-wide data base in which both Vault and Quick Books data are no longer separate, but now represent the same item. Vault API modification would be necessary to push/pull data from database. Take it with a grain of salt, but I would venture to say that the method of linking Quick Books to a data base would be far less in depth than modifying Vaults API based on my past experience working with Quick Books.

There would be a decent amount of work required to implement this method. For experimental purposes Microsoft SQL Server Express (free for small commercial applications) data base system could be utilized. Should we out-grow this database, SQL server is extremely scalable, and using the Express version of the platform as a starting point would be a good option.

Without full scale implementation and debugging the two proposed solutions, my opinion is this:

Saving and storing the Excel BOM on Vault addresses the CAPA immediately, and requires would require only a small amount of effort. However, it still leaves the door open to possible errors.

I do not feel that continuing to populate an Excel document with data that is neither common, nor dynamic is beneficial to Stella Maris. There are instances when time is not an ally to producing a bid, or making a production deadline. It is in these instances that access to accurate, timely data to make decisions are of utmost importance. As it currently stands you could say that there are four “databases” used to retrieve data: Vault, Quick Books, historical Excel BOMs (X: Drive), and the internet. By internet I mean turning to a manufacturer website to retrieve data and drawings before looking in our records for items we have used in the past. In the following sections my intention is to not only address the CAPA, but also address what I believe is the reason for its existence.

INVENTOR BOM

I strongly believe that Engineering, Drafting, Project Management, and the Accounting Departments should all be looking at the same single instance of product data, company-wide. Without commonality, you will find redundant, and often times inaccurate information. The “final” solution(s) that I am proposing would eliminate “static data” Excel BOMs, and reduce the amount of time spent on estimating, designing, and management of the project. It eliminates redundant and inaccurate data, and would offers scalability. One of these two “final” solutions retain a lot of current Stella Maris operating procedures. The other, although complete and thorough, would require a not only a significant amount of time and resources to implement, but also maintain.

#### OBSERVATIONS

I will briefly (and roughly) describe the creation and structure of the 3D and 2D drawing files. In addition, I will describe the initial work flow between Mechanical Engineering and Drafting during the design phase.

Each .ipt or .iam file contains user specified non-CAD related data, such as vendor, vendor part number, description, pricing, leadtime, etc. These non-CAD data fields are THE most important source of information to be able to use the Inventor BOM for its true and a resourceful purpose. However, while Drafting does make an effort to keep the non-CAD related data about the drawing as up-to-date as possible, it takes on average, about thirty (30) minutes of their time modify the non-CAD data properties of that particular 500xxx part. They then would have to spend further time ensuring the non-CAD data matches throughout the rest of the drawing package. Furthermore, if at a later date, the same 3D or 2D drawing was used on another project, and data such as vendor or pricing was updated to current information, it would change the data in all other instances where this drawing was used and thus change the BOM for those projects

This is why Drafting usually only populates critical, non-CAD data and leaves the rest alone. Another reason for leaving the other data fields alone is that often times, drawings for gauges, valves, etc. can be “reused” for other projects. There is also the case where the drawing does not make it to Drafting in time to generate a 2D or 3D model of the part. In this situation, Drafting is forced to just “wing-it” as best they can from the data sheet to the part. This promotes a more rapid production of a 2D or 3D drawing, but is the single reason why the Excel BOM exists, and the reason why we cannot use the Inventor generated BOM.

The following is an example of the common workflow between Mechanical Engineering and the Drafting Department:

1. Engineering compiles a “static” Excel BOM and assigns a sequential number to each of the spec’d products that are listed. Concurrently, a hand sketched flow diagram is generated. Both of these documents are then sent to Drafting via email or X: Drive
2. Drafting receives both documents, and from the hand sketch, creates a 2D flow diagram. Drafting then copies over the Excel BOM data into a table and places it on the 2D flow diagram.
3. From the sequential item numbers specified by Engineering in the Excel BOM, Drafting tags each corresponding part on the flow diagram with a matching number item balloon. It is important to remember that at this point the 2D or 3D part that was used is either just a “placeholder” part
4. This document is then sent back to Engineering, and upon receipt, Engineering must go through and ensure that every item that was specified in the Excel BOM was included on the 2D flow diagram. Engineering must also check to see if a “placeholder” drawing was used. If so then it is critical that they go back and find the actual drawing of the part, whether it is already in Vault, online, or emailed from the vendor, to make sure that the “placeholder” geometry matches the geometry of the actual part. During my interviews, I found that very few Engineers were looking in Vault for the corresponding drawing BEFORE reaching out to vendors. I am assuming all of this stems from the fact that Stella Maris rarely makes the same product more than once. In other instances, the vendor supplied drawing is given to Drafting, but file formatting issues have occurred where that drawing could not be used, and therefore other methods must be used to generate the drawing.
5. Once Drafting finishes the assembly, a 2D representation of the 3D model is generated in an .idw file format. Drafting will go into this 2D .idw and populate it with the same parts list based on the Excel BOM, and repeat the steps to assign balloons to the corresponding items.

It was attempted some time ago, to bring the Inventor BOM to full accuracy, with the exception of miscellaneous non-CAD items. For every Excel BOM part, the correct and accurate 2D .dwg or 3D .ipt was used. The data associated with each part was then entered into the correct property field either from Inventor itself, or within Vault. Each part was sequentially inserted into the .iam assembly to match the Excel BOM sequential numbering generated by Engineering. This resulted in an extraordinary amount of effort and time spent to ensure the accuracy of the Inventor BOM.

The biggest time consumer of this attempt was when Project Management or Manufacturing needed to change a part after the release of the drawing. This situation usually stems from either the discovery of unacceptable part lead times on the part something not being caught and addressed by Engineering before the BOM was released to Project Management. Upon agreement that the change needed to be made, Drafting removed and replaced the part, which subsequently renumbered all items in the Inventor BOM. This meant that the Inventor BOM no longer matched the 2D .idw parts list, nor the Excel BOM, nor the 2D .dwg flow diagram. To remedy this, Drafting pulled each part out one by one, and re-inserted until the sequential numbering in the Inventor BOM matched the other 2D/Excel BOM item numbers. After this attempt, creating an accurate BOM within Inventor was never tried again.

#### Summary

I believe that this effort was a step in the right direction, but it did not use Inventor BOM in the sense that it should be. Upon creation of the project .iam assembly, once the Inventor BOM is accurate and filled with the information matching the Engineering spec’d parts (not from an Excel BOM, ill go into detail later), no further modification of the Inventor BOM is necessary. The BOM should be released as is. Any further modification of a part that does not change the overall fit, form, or function should be done outside of Inventor, usually within an ERP system. To be more specific, the Inventor BOM should be exported into a company-wide data base. With a shared data base, the parts that are populated on the Inventor BOM will be the same exact matching part listed within the database that is viewed by the rest of the company. A modification within the ERP would in no way effect any geometry of the drawing file. It can however populate critical non-CAD data to the drawing file. If a part substitution is necessary that changes the fit, form or function, the users (SM Project Management) of the ERP can be prompted to select from a drop down list containing parts that fall within the Design Acceptance Criteria (I will explain how this will work for our situation later).

I use the example of how this could work with an ERP, but by no means am I suggesting that an ERP is necessary at this point in time for Stella Maris. ERPs are extremely costly to implement and can cause more trouble than good if implemented incorrectly. From my past experience working for a similar “Custom-Engineer-To-Order” company, the ERP system eliminated a lot of the issues that are presented in this document. The difference between Stella Maris and the other company is that there was inventory to account for, as well as a service department that sold parts. Tracking of inventory was done by hand, which delayed the system and caused numerous problems. I would also say that only about 15% of the ERPs potential was used, most of which was strictly for Accounting.

# final Solution

The “ultimate” solution here would be to eliminate using any “placeholder” drawing files, and use files that represent the actual part. Again, a great deal of time, effort and investment would be necessary to implement this. This method would still require the deployment of a company-wide database, and more than likely, the use of an ERP system. The solution that I think would best fit Stella Maris consists of a combination of the following methods:

There are several Vault add-ons developed by a company named Cool Orange. This business is heavily involved with integrating Vault into various business systems. One of their product offerings merges Vault data with an ERP system instantaneously. I have attached a link to the video showing the demonstration The ERP system deployed in their demonstration is Microsoft Dynamics NAV. From this demonstration you can see that it is possible to assign an item that lives in the ERP database, to the corresponding Vault .ipt file. In the same manner a project listed in the ERP can be linked to the .iam assembly in Vault.

A “DAC” list of approved vendors and substitutes can be created and linked directly to the part file. This can be done from within both Vault, and the ERP system allowing the user to select pre-approved parts. This can be very powerful from an Estimating, Engineering, Drafting and Project Management point of view.

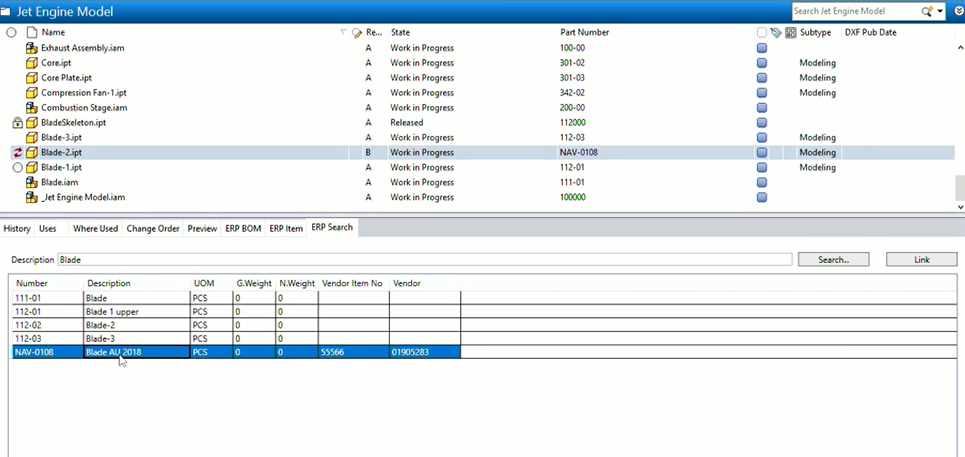
The program developed by Cool Orange exposes some of Vault API customization capabilities. While I do not believe that Stella Maris needs an ERP system at this point in time, my research has led me to believe that we can set the ground work to prepare ourselves for one, while at the same time utilizing this base system (database) for our needs until we finally grow large enough to need the ERP.

Cool Orange modifies Vault’s API code to push data into a database used by the ERP system. My suggestion is that we look into either using Cool Orange to push and pull data between Vault, and an SQL database, or write our own program. From what I have gathered this looks to be absolutely possible utilizing a PowerShell script.

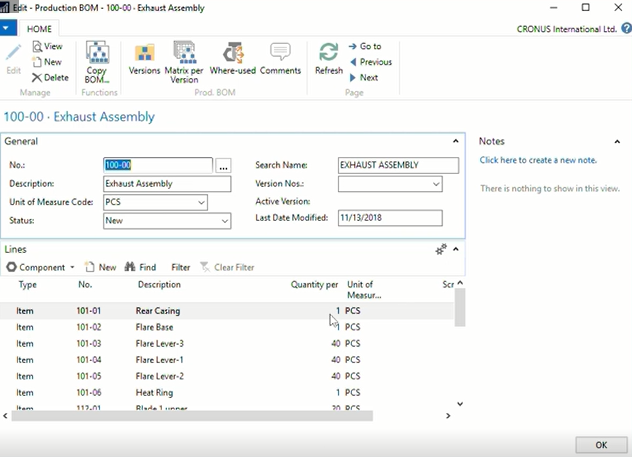
Not to ignore the capabilities of Microsoft Excel, Engineering could still use Excel for its ease-of-use to create a BOM. As an example of an application, with an SQL database deployed, Estimating could gather historical data much faster than current methods with Engineering and Quick Books data being one in the same. But in this example the Excel BOM would be tied to the SQL database. Creation of an Excel BOM could only be done within Vault, forcing a standard template and rules. The information that is pulled into Excel can be dynamic and live, and would eliminate the need to send the BOM away from the local computer. Drafting can also have an Excel document designed to retrieve this live data and display it in an efficient manner. Drafting could still use their “placeholder” drawings when necessary, and through the Cool Orange integration, the relevant item data could be linked to that placeholder drawing. Miscellaneous items could be added as scratch items. This would create a completely accurate BOM that can be exported back into the database for Project Management to use. As an additional option that avoids the need for an ERP, a GUI can be built from Visual Studio for each department that would query the SQL server for such information, and push it back when necessary.

I do want to express that I fully understand that I have played absolutely no role in bringing this business to where it is today. From my past work experience, I have a developed a deep appreciation of just how much thought, effort, and if I may, trust in the good Lord above it takes to keep the business not only running, but thriving. In no way shape or form am I attacking “what works”. I am only using what I know from previous experience, coupled with observations and research to provide the best opinion I can possibly give.

There are many solutions at our fingertips to address BOM revision control, and I’ve only really touched on a handful of them. While it is not my decision to choose which path we take, I hope this helps in making that determination.

*Cool Orange - Vault to ERP*

*Cool Orange - ERP to Vault*



# Resources

Video of Cool Orange Demonstration -

X:\Personal Folders\Jordan Landry\Resources

*Cool Orange*

https://www.coolorange.com/en-us/index.html

*Quiring SQL Server*

<https://support.coolorange.com/support/solutions/articles/22000216551-query-an-sql-server>

Cool Orange powerGate

<https://www.coolorange.com/en-us/connect.html#powerGate>

Cool Orange powerVault

https://www.coolorange.com/en-us/vault.html